

# Making Point Counting Work for You

**Presented by: Donna Szakal**  
***Software Process Improvement (SPI) Project***

# Purpose and Objectives

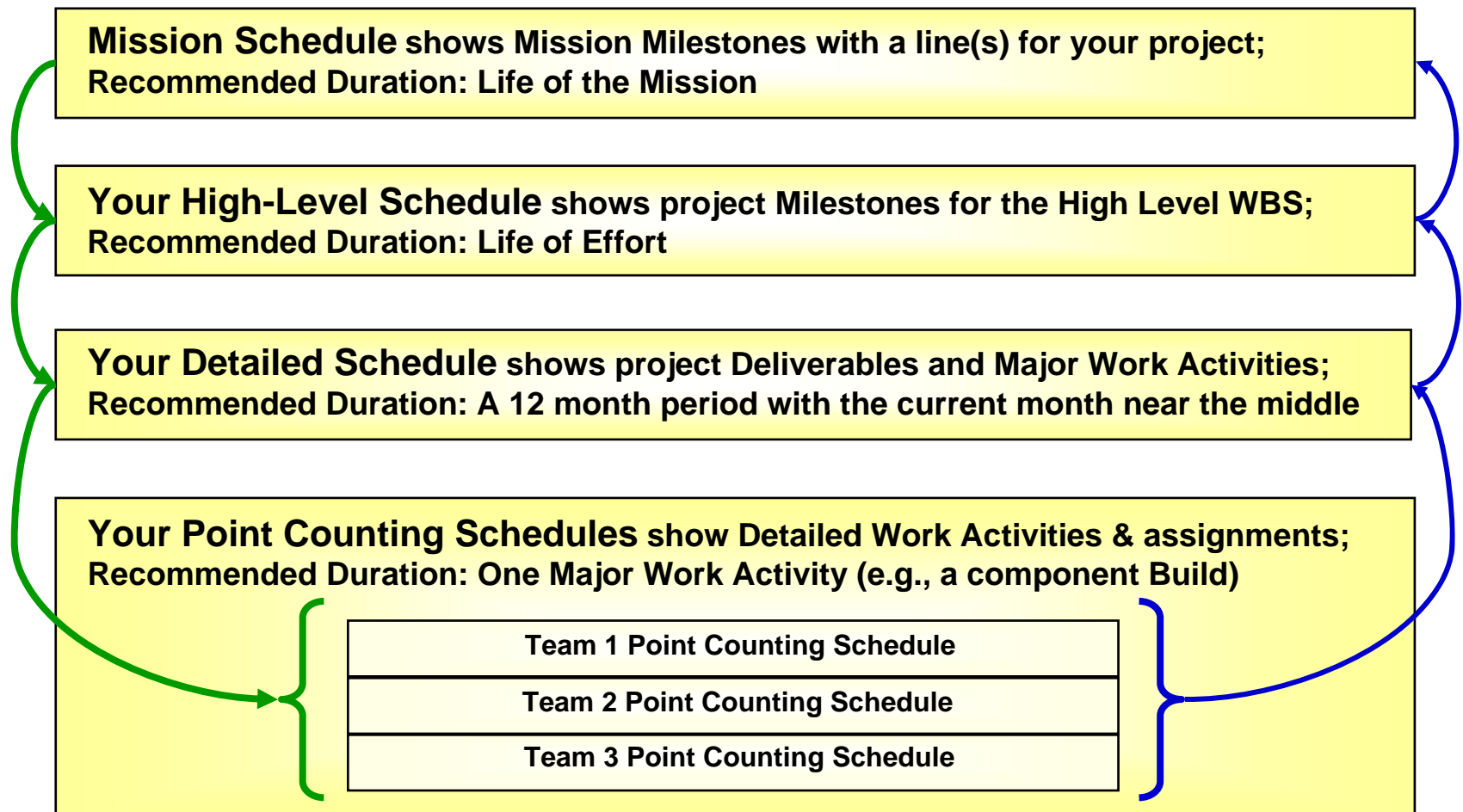
- **Purpose: To acquaint you with point counting and how it can be applied to your project**
- **Objective: Help you understand:**
  - **How point counting can assist you in schedule progress tracking**
  - **How to create a point counting plan**
  - **How to monitor progress using point counting**
  - **How to report progress tracked with point counting**

## Understanding the Starting Point ...

# Where Point Counting Fits In

Plan major activities based on higher-level schedules and work down as detail is added

Collect status at lower levels and roll it up

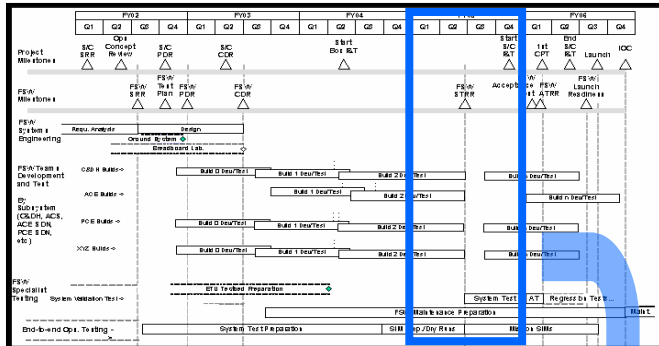


# Getting to the Point Counting Schedules

- 3 Subsystem Development**
  - 3.1 Architecture Design**
    - 3.1.1 Define Requirements**
      - 3.1.1.1 Meet Stakeholders to Understand L3 Req
      - 3.1.1.2 Analyze Level 3 Req
      - 3.1.1.3 Allocate L3 Reqs to Subsystems
      - 3.1.1.4 Baseline Level 3 Requirements
      - 3.1.1.5 Conduct Software Req Review
      - 3.1.1.6 Capture and Track SRR RFAs
    - 3.1.2 Develop Preliminary Design Pkg**
      - 3.1.2.1 Conduct Prelim Design Review
      - 3.1.2.2 Capture and Track PDR RFAs
    - 3.1.3 Develop Detailed Design Pkg**
      - 3.1.3.1 Conduct Crit Design Review
      - 3.1.3.2 Capture and Track CDR RFAs
  - 3.2 Develop Software**
    - 3.2.1 Develop C&DH Level 4 Requirements**
    - 3.2.2 Develop C&DH Software**
      - 3.2.2.1 PSE Build 1.0**
        - 3.2.2.1.1 Design and Develop Build**
        - 3.2.2.1.2 Perform Build Integration Test**
      - 3.2.2.2 PSE Build 2.0**
        - 3.2.2.2.1 Design and Develop Build
        - 3.2.2.2.2 Perform Build Integration Test
      - 3.2.2.3 PSE Build 3.0**
        - 3.2.2.3.1 Design and Develop Build
        - 3.2.2.3.2 Perform Build Integration Test
      - 3.2.2.4 Create/Update PSE User's Guide**
      - 3.2.2.5 S/COMM Build 1.0**
        - 3.2.2.5.1 Design and Develop Build
        - 3.2.2.5.2 Perform Build Integration Test
      - 3.2.2.6 S/COMM Build 2.0**
        - 3.2.2.6.1 Design and Develop Build

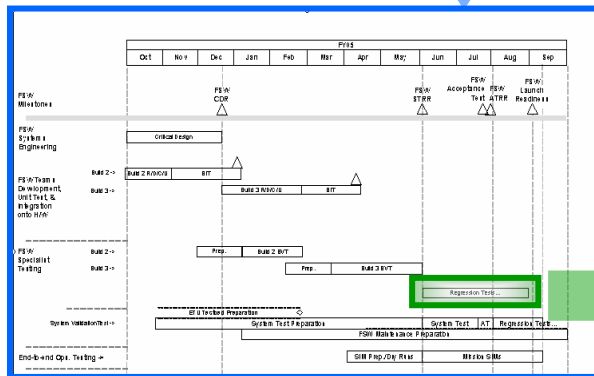
3.2	Develop Software						
3.2.1	Develop C&DH Level 4 Requirements						
3.2.2	Develop C&DH Software						
3.2.2.1	PSE Build 1					435	850

Group	Task Name	Points	Planned Start	Planned Finish	Actual Finish	Sub-system	Assigned to	Status	02/03/06	02/10/06	02/17/06	02/24/06	03/03/06	03/10/06
0	SubSys1		NA	NA	NA									
0	John SubSys1 Component 1	25	02/03/06	03/10/06	NA	ss1	John	Late						due
	John SubSys1 Component 2	10	02/23/06	03/10/06	03/10/06	ss1	John	10.0						10.0
	John SubSys1 Component 3	30	02/28/06	04/07/06	04/07/06	ss1	John	30.0						
	Paul SubSys1 Component 4	25	03/08/06	04/11/06	NA	ss1	Paul	Late						
	Paul SubSys1 Component 5	15	03/20/06	04/07/06	04/12/06	ss1	Paul	15.0						
	Paul SubSys1 Component 6	10	02/07/06	02/22/06	NA	ss1	Paul	Late				due		



## Multi-Year High Level Schedule

- Cradle-to-Grave Mission Project-Level Schedule
- Major Mission Project Milestones



## Current Year Detailed Schedule

- One Year Team-Level Schedule
- Major Activity-Level Milestones

[illegible]

## Point Counting Schedule

- Activity-Level Detailed Schedule
- Point Counting

# What is Point Counting ?

- **Method used to**
  - Plan work activities at the lowest level of the Work Breakdown Structure (WBS)
  - Assign an effort (i.e., “points”) to each of those activities
  - Earn credit (i.e., “points”) for accomplishing those activities
- **Also referred to as Progress Tracking**
- **Based on the Earned Value method**
- **“Equivalent Units”**

# Point Counting Planning

# Planning a Point Counting Schedule

- 1. Select appropriate, milestone-based detailed schedule components**
- 2. Partition components into lower level detailed activities**
- 3. Assign an objective weight to each activity**
- 4. Assign an individual to each activity**
- 5. Schedule each activity**

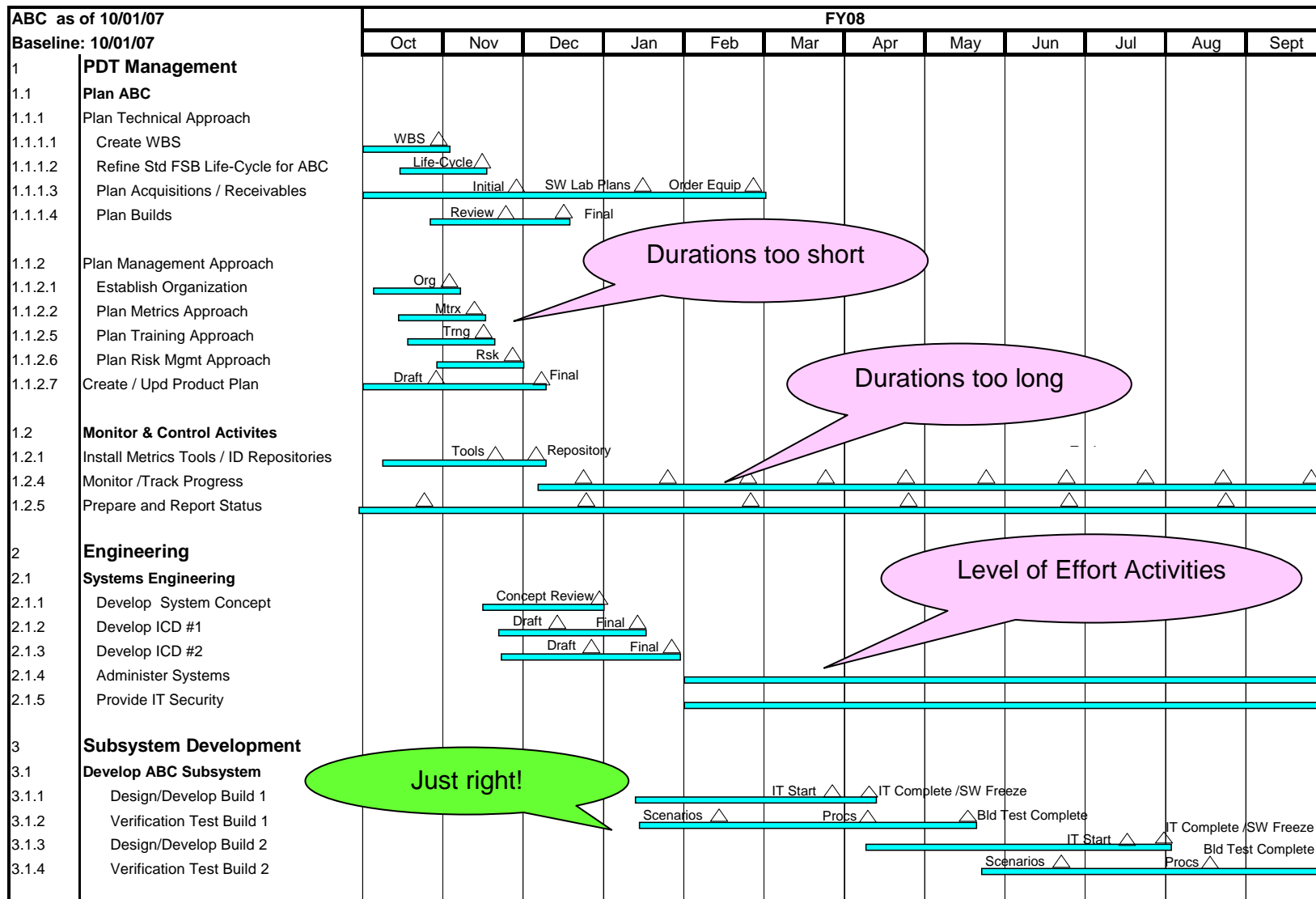
# Planning: 1. Selecting appropriate, milestone-based ...

- **“Appropriate” components are those composed of**
  - A set of similar activities
  - Performed within the same period of time
  - Performed over a period of ~3 or more months
  - Contribute to a single interim or end product
- **Examples:**
  - Implementation of a single Build/Release
  - “Independent” testing of a single Build/Release
  - System testing of a single Build/Release
- **In general, “appropriate” components are NOT “level of effort” components such as ongoing system administration, etc.**

## *Hint:*

- *Selecting too many components may lead to bookkeeping overkill*
- *Selecting too few components may hide schedule problems you would otherwise find using the point counting approach*
- *Selecting components of 2 months or less in duration does not add sufficient value*

# What WBS Elements are Appropriate for Point Counting?



# Planning: 2. Partition components into lower level detailed activities ...

- Partition components into detailed activities that
  - Provide insight into schedule problems as early as possible
  - Have objective criteria that define activity completion (e.g., peer review held, test results recorded)
- “Traditional” partitioning includes:
  - Implementation of a single Build/Release (new development or maintenance) into:
    - Unit Design
    - Unit Code
    - Unit Test
  - “Independent” or system testing of a single Build/Release into:
    - Writing test scenarios / procedures
    - Generating test data
    - Conducting tests

## *Hint:*

- *Finding the right component partitioning is a matter of experience (use Unit Design, Unit Code, and Unit Test rather than Unit Start, Unit End)*
- *Use records routinely created in the course of business to define activity completion*

# Planning: 3. Assigning objective weights ...

- **Point counting assumes a rationale for objectively weighting activities**
  - Use historical data, models, feedback from staff, your BOE to make reasonable estimates of effort for each activity
- **Define a point counting scheme that reflects effort, for example:**
  - Assign 1 point to a 1 hour effort
  - Assign points to sets of activities for which you have a feel for the relative effort of each to the other: 5 points for design, 2 for code, and 3 for unit test (total of 10 points per unit)

## *Hint:*

- *Always review points weighting with people doing the work to obtain buy-in*

# Planning: 4. Assigning an individual to each activity ...

- **Ensure there is warm body to assign to perform each activity at its scheduled start!**
  - **If no one is assigned ...  
you'll see no progress ...  
and you'll earn no points !**

## Planning: 5. Scheduling each activity ...

- **Order activities:**
  - In their natural sequence (e.g., design, code, test)
  - Based on dependencies among them
  - As the person doing the work plans to work on them

*Note: Points are earned as activities are complete regardless of ordering*
- **Schedule activities within the given constraints**
  - Allow sufficient schedule time to cover the activity
  - Don't overbook the assigned individuals
  - Always schedule activities to start and end during the period covered by point counting
- **Ask the person who will do the work to schedule their assigned activities, then review their schedule carefully**
  - “People tend to support what they help to create”

### *Hint:*

- *Don't fall into the trap of thinking that 8 hours of effort = one day of time !*
- *Know whether the individual who scheduled the activity(s) is an optimist or a pessimist and adjust accordingly (with their concurrence, of course)!*

# A Simple Example: The Plan

As of Date 02/03/07			Sub-Plan			0	0	0	0	0	0	0	0	0	0	0	0	0			
Project Start Date 01/27/07			Sub-Actual			0	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A			
Plan			0			10	12	27	30	34	34	40	40	40	40	50	50	50			
Task Room 15			Actual			0	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A			
Sub-Group			Baseline			50	50	50	50	50	50	50	50	50	50	50	50	50			
Filter # 2			Upper			0	0	13	16	35	39	44	44	50	50	50	50	50			
Col. G			Lower			0	0	0	1	20	24	29	29	37	37	37	37	50			
Filter 1 G			Earliest			Plan Rate			0	0	10	2	15	3	4	0	6	0	0	0	10
Filter 2 H			02/03/07 02/17/07 01/00/00			Actual Rate			0	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
			Latest			Av PI Rt			0	0	3	3	5	5	4	4	4	4	3	4	
			04/07/07 04/23/07 01/00/00			Av Ac Rt			0	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	
						Days			7	14	21	28	35	42	49	56	63	70	77	84	91
Group	Task Name	Points	Planned Start	Planned Finish	Actual Finish	Status	02/03/07	02/10/07	02/17/07	02/24/07	03/03/07	03/10/07	03/17/07	03/24/07	03/31/07	04/07/07	04/14/07	04/21/07	04/28/07		
0	Component 1	NA	NA	NA																	
John	Unit Design	5	02/03/07	02/17/07	NA				due												
John	Unit Code	2	02/18/07	02/24/07	NA					due											
John	Unit Test	3	02/24/07	03/03/07	NA						due										
0	Component 2	NA	NA	NA																	
Paul	Unit Design	5	02/03/07	02/17/07	NA				due												
Paul	Unit Code	2	02/18/07	03/01/07	NA					due											
Paul	Unit Test	3	03/02/07	03/10/07	NA						due										
0	Component 3	NA	NA	NA																	
George	Unit Design	10	02/10/07	03/01/07	NA					due											
George	Unit Code	4	03/02/07	03/11/07	NA						due										
George	Unit Test	6	03/12/07	03/25/07	NA							due									
Ringo	Integration Testing	10	04/07/07	04/23/07	NA									due							
0		NA	NA	NA																	

## Planning

- Assign an “effort” (also known as points) to each activity
- Assign “Planned Start”, “Planned Finish” dates for each activity
- Columns represent weeks
- “Due” shows planned week of completion
- Cumulative planned points show on “Plan” line (yellow); Baseline points show on “Baseline” line (blue)

# Monitoring and Controlling With Point Counting

# Monitoring and Control of Point Counting Schedules

- **Collect completion information on a regular basis**
  - Weekly is good, a minimum is every 2 weeks (e.g., “every other Friday”)
  - Earn credit as soon as an activity is completed
- **Compare cumulative planned points to the cumulative completed (i.e., performed) points**
- **Analyze variances from the plan**
- **Identify developing trends and take corrective action before they become serious**

## *Hint:*

- *Collecting completion information too frequently can be a bookkeeping nightmare; too infrequently can delay finding and resolving problems*
- *Persevere in using the technique. Approaching deadlines can tempt a project to stop tracking. Continue tracking and responding to variances with corrective action*

# A Simple Example Tracking

As of Date 03/03/07			Sub-Plan			0	0	0	0	0	0	0	0	0	0	0	0	0			
Project Start Date 01/27/07			Sub-Actual			0	0	0	0	0	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A			
			Plan			0	0	10	12	27	30	34	34	40	40	40	40	50			
<div>Task Room15</div>			Actual			0	0	10	12	22	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A			
<div>Sub-Group</div>			Baseline			50	50	50	50	50	50	50	50	50	50	50	50	50			
<div>Filter #2</div>			Upper			0	0	13	16	35	39	44	44	50	50	50	50	50			
			Lower			0	0	0	1	20	24	29	29	37	37	37	37	50			
Filter 1 G			Earliest			Plan Rate			0	0	10	2	15	3	4	0	6	0	0	10	
Filter 2 H			02/03/07 02/17/07 02/17/07			Actual Rate			0	0	10	2	10	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	
			Latest			Av PI Rt			0	0	3	3	5	5	5	4	4	4	3	4	
			04/07/07 04/23/07 03/02/07			Av Ac Rt			0	0	3	3	4	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	
						Days			7	14	21	28	35	42	49	56	63	70	77	84	91
Group	Task Name	Points	Planned Start	Planned Finish	Actual Finish	Status	02/03/07	02/10/07	02/17/07	02/24/07	03/03/07	03/10/07	03/17/07	03/24/07	03/31/07	04/07/07	04/14/07	04/21/07	04/28/07		
0			NA	NA	NA																
0	Component 1		NA	NA	NA																
John	Unit Design	5	02/03/07	02/17/07	02/17/07	5.0			5.0												
John	Unit Code	2	02/18/07	02/24/07	02/24/07	2.0				2.0											
John	Unit Test	3	02/24/07	03/03/07	NA	Late					due										
0	Component 2		NA	NA	NA																
Paul	Unit Design	5	02/03/07	02/17/07	02/17/07	5.0			5.0												
Paul	Unit Code	2	02/18/07	03/01/07	NA	Late					due										
Paul	Unit Test	3	03/02/07	03/10/07	NA							due									
0	Component 3		NA	NA	NA																
George	Unit Design	10	02/10/07	03/01/07	03/02/07	10.0					10.0										
George	Unit Code	4	03/02/07	03/11/07	NA							due									
George	Unit Test	6	03/12/07	03/25/07	NA									due							
Ringo	Integration Testing	10	04/07/07	04/23/07	NA															due	
0			NA	NA	NA																

## Tracking

- Fill in “Actual Finish” date for completed activities
- Number indicates points earned for activity
- “Late” indicates activity is not completed as scheduled
- Cumulative points earned to date show on “Actual” line (pink)

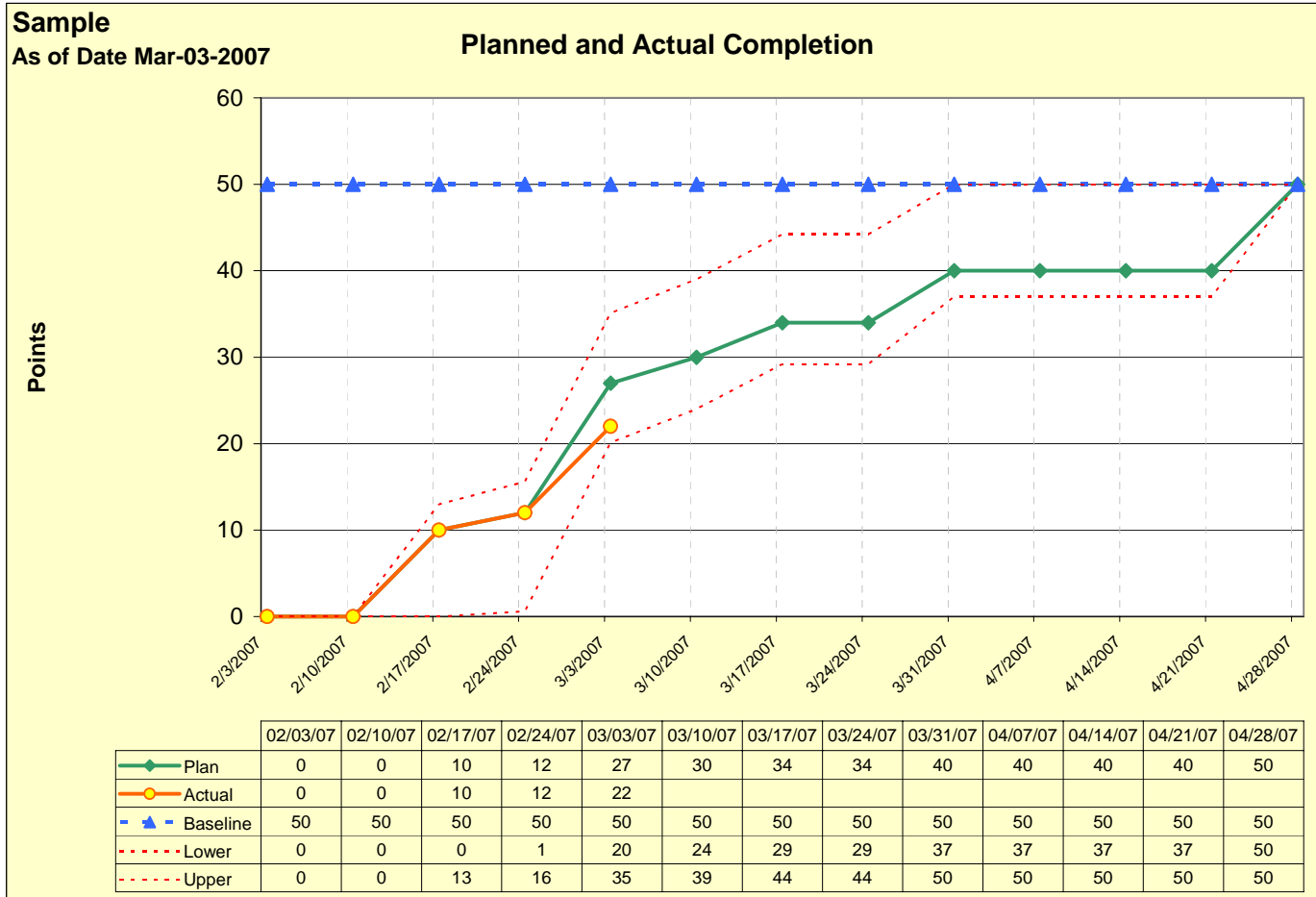
# Reporting Point Counting Schedules

- **Use the latest Point Counting file charts to populate your Branch Status Review (BSR)**
- **Summarize and document your analysis:**
  - **Analysis: Reason(s) for actuals varying from the plan**
  - **Impact: Impact(s) of the variance**
  - **Corrective Actions: Actions planned or taken to resolve the variance (...and track these actions to closure!)**

## *Hint:*

- *Reporting schedules should not drive collection and analysis of Point Counting data*

# A Simple Example Trend Chart



**Analysis:** Build development is falling behind schedule due to greater than expected complexity in several units.  
**Impact:** Delivery to the test team expected to be one week late.  
**Corrective Action:** Expedite planned meetings with Code 000 to seek ways to reduce complexity of the interface.

# Example Reasons for Variances in Point Counting Schedules

- Unexpected problem, complexity, scope (size), extent of specific work activities
- Unplanned but necessary work activities performed (specify)
- Incomplete or missing equipment, software, or information (e.g., items supplied by others or purchased items not yet available)
- Change in requirements
- Unexpectedly low productivity due to learning curve/need for training
- Staffing shortage or staffing skill mismatch

# Example Corrective Actions

- Use additional staff to ... (but remember this might mean a cost impact)
- Use parallel workarounds (specify)
- Seek simpler/faster solutions in another (specific) area of the project
- Improve productivity by ...
- Recover the point counting earning lag by ...
- Modify the plan via an authorized change
- None (e.g., condition cannot be corrected, variance is unrecoverable, trend is expected to continue)

## Other Odds and Ends

# Point Counting Tool\*

- **Point Counting Tool\*** is an Excel spreadsheet that allows projects to plan and track the progress of detailed schedule activities
  - Find it at <http://software.gsfc.nasa.gov/tools.cfm>
  - It supports planning, monitoring and control, and reporting
  - It contains trend information to provide insight into progress including the ability to meet schedules
- **Detailed User's Guide** is a separate Word file
- **Two versions of the tool:**
  - Single Activity spreadsheet
  - Multiple Activity spreadsheet

## *Hint:*

- *Unless you have a significant amount of experience in using the point counting approach, use the Single Activity spreadsheet*
- *Use a file naming convention that includes the date of update*

# Point Counting Pitfalls

- **Confusing effort (i.e., points) with time (i.e., schedule)**
  - Don't plan an 8 hour effort to be completed in one day
- **Combining unrelated activities in one point counting file**
  - Don't include writing documents in point counting files containing development or test activities
  - Don't combine development and independent test activities
  - Don't combine builds
- **Planning so that most of the points are earned during the last few weeks of the effort without the benefit of additional staff**
  - Do you plan for people to work harder / faster?
- **Planning flimsy “proof” of accomplishments**
  - Use “proof” (e.g., unit design walkthroughs) vs testimonials (e.g., “I finished the design”)

# Records You Should Keep

- **Sequence of dated copies of the Point Counting Tool Files (every update)**
  - Initial copy showing only the initial Baseline Plan
  - Subsequent copies showing plan with associated actual progress (each copy shows incremental updates)
  - Each copy shows that incremental updates are being made
- **Sequence of dated copies of the Point Counting Analysis (every reporting period)**
  - Analysis, Impact, and Corrective Actions
  - Demonstrates periodic assessment and review of progress
- **Records of corrective actions taken and tracked to closure**

# Point Counting Summary

- Provides an objective measure of progress against the schedule
- Enables detection of the need for corrective action in time to prevent a problem or minimize its impact
- Improves ability to estimate completion costs and schedule variances by analysis of accumulated data and trends
- Provides an objectivity that is often difficult to maintain during the heat of crisis

# *Questions?*

# Acronyms

- **BSR – Branch Status Review**
- **SPI – Software Process Improvement**
- **WBS – Work Breakdown Structure**